**Part 1: Yelp Dataset Profiling and Understanding**

1. Profile the data by finding the total number of records for each of the tables below:

i. Attribute table = 10000

ii. Business table = 10000

iii. Category table = 10000

iv. Checkin table = 10000

v. elite\_years table =10000

vi. friend table = 10000

vii. hours table = 10000

viii. photo table = 10000

ix. review table = 10000

x. tip table = 10000

xi. user table = 10000

Code:

SELECT

(SELECT COUNT(\*)FROM Attribute) AS Attribute\_count,

(SELECT COUNT(\*)FROM Business) AS Business\_count,

(SELECT COUNT(\*)FROM Category) AS Category\_count,

(SELECT COUNT(\*)FROM Checkin) AS Checkin\_count,

(SELECT COUNT(\*)FROM elite\_years) AS elite\_years\_count,

(SELECT COUNT(\*)FROM friend) AS friend\_count,

(SELECT COUNT(\*)FROM hours) AS hours\_count,

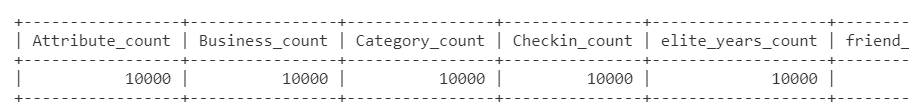
(SELECT COUNT(\*)FROM photo) AS photo\_count,

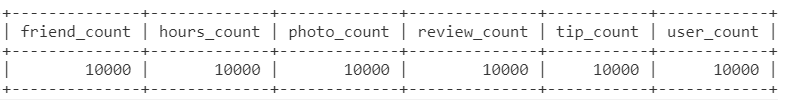
(SELECT COUNT(\*)FROM review) AS review\_count,

(SELECT COUNT(\*)FROM tip) AS tip\_count,

(SELECT COUNT(\*)FROM user) AS user\_count;

Output:





2. Find the total distinct records by either the foreign key or primary key for each table. If two foreign keys are listed in the table, please specify which foreign key.

i. Business = 10000

ii. Hours = 1562

iii. Category = 2643

iv. Attribute = 1115

v. Review = 10000

vi. Checkin = 493

vii. Photo = 10000

viii. Tip = 537

ix. User = 10000

x. Friend = 11

xi. Elite\_years = 2780

Note: Primary Keys are denoted in the ER-Diagram with a yellow key icon.

Code:

SELECT

(SELECT COUNT(DISTINCT (id))FROM Business) AS Business,

(SELECT COUNT(DISTINCT (business\_id))FROM Hours) AS Hours,

(SELECT COUNT(DISTINCT (business\_id))FROM Category) AS Category,

(SELECT COUNT(DISTINCT (business\_id))FROM Attribute) AS Attribute,

(SELECT COUNT(DISTINCT (id))FROM Review) AS Review,

(SELECT COUNT(DISTINCT (business\_id))FROM Checkin) AS Checkin,

(SELECT COUNT(DISTINCT (id))FROM Photo) AS Photo,

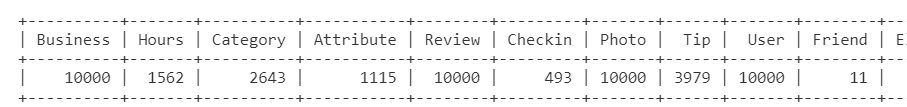
(SELECT COUNT(DISTINCT (user\_id))FROM Tip) AS Tip,

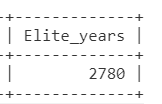
(SELECT COUNT(DISTINCT (id))FROM User) AS User,

(SELECT COUNT(DISTINCT (user\_id))FROM Friend) AS Friend,

(SELECT COUNT(DISTINCT (user\_id))FROM Elite\_years) AS Elite\_years;

Output:





3. Are there any columns with null values in the Users table? Indicate "yes," or "no."

Answer: NO

SQL code used to arrive at answer:

Code:

SELECT COUNT(id)

        , COUNT(name)

        , COUNT(review\_count)

        , COUNT(yelping\_since)

        , COUNT(useful)

        , COUNT(funny)

        , COUNT(cool)

        , COUNT(fans)

        , COUNT(average\_stars)

        , COUNT(compliment\_hot)

        , COUNT(compliment\_more)

        , COUNT(compliment\_profile)

        , COUNT(compliment\_cute)

        , COUNT(compliment\_list)

        , COUNT(compliment\_note)

        , COUNT(compliment\_plain)

        , COUNT(compliment\_cool)

        , COUNT(compliment\_funny)

        , COUNT(compliment\_writer)

        , COUNT(compliment\_photos)

FROM User

WHERE id IS NULL

        OR review\_count IS NULL

        OR yelping\_since IS NULL

        OR useful IS NULL

        OR funny IS NULL

        OR cool IS NULL

        OR fans IS NULL

        OR average\_stars IS NULL

        OR compliment\_hot IS NULL

        OR compliment\_more IS NULL

        OR compliment\_profile IS NULL

        OR compliment\_cute IS NULL

        OR compliment\_list IS NULL

        OR compliment\_note IS NULL

        OR compliment\_plain IS NULL

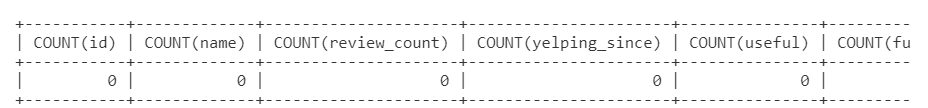
        OR compliment\_cool IS NULL

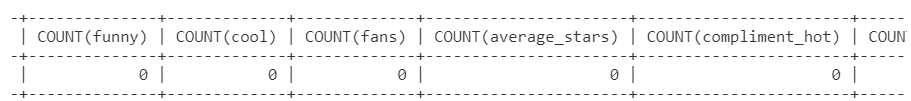
        OR compliment\_funny IS NULL

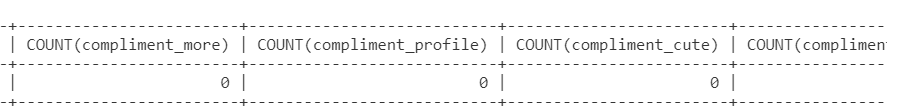
        OR compliment\_writer IS NULL

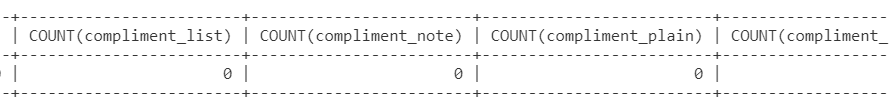
        OR compliment\_photos IS NULL

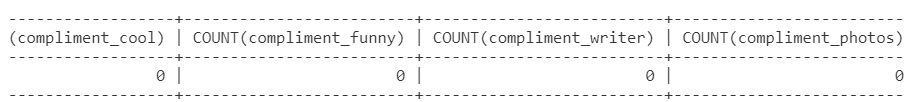
Output:











4. For each table and column listed below, display the smallest (minimum), largest (maximum), and average (mean) value for the following fields:

i. Table: Review, Column: Stars

min:1 max:5 avg:3.7082

ii. Table: Business, Column: Stars

min:1.0 max: 5.0 avg:3.6549

iii. Table: Tip, Column: Likes

min:0 max:2 avg:0.0144

iv. Table: Checkin, Column: Count

min:1 max:53 avg:1.9414

v. Table: User, Column: Review\_count

min:0 max:2000 avg:24.2995

Code:

SELECT

(SELECT MIN(stars)FROM review) AS Review,

(SELECT MIN(stars)FROM Business) AS Business,

(SELECT MIN(likes)FROM Tip) AS Tip,

(SELECT MIN(Count)FROM Checkin) AS Checkin,

(SELECT MIN(Review\_count)FROM User) AS User

UNION

SELECT

(SELECT MAX(stars)FROM review),

(SELECT MAX(stars)FROM Business),

(SELECT MAX(likes)FROM Tip),

(SELECT MAX(Count)FROM Checkin) ,

(SELECT MAX(Review\_count)FROM User)

UNION

SELECT

(SELECT AVG(stars)FROM review),

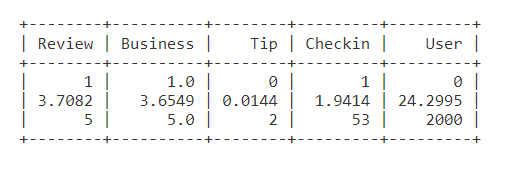
(SELECT AVG(stars)FROM Business),

(SELECT AVG(likes)FROM Tip) ,

(SELECT AVG(Count)FROM Checkin),

(SELECT AVG(Review\_count)FROM User);

Output:



Alternative Code:

SELECT MIN(COALESCE(STARS,0)) AS MIN,MAX(COALESCE(STARS,0)) AS MAX,AVG(COALESCE(STARS,0)) AS AVG FROM REVIEW

UNION ALL

 SELECT MIN(COALESCE(STARS,0)) AS MIN, MAX(COALESCE(STARS,0))AS MAX,AVG(COALESCE(STARS,0))AS AVG  FROM BUSINESS

UNION ALL

 SELECT MIN(COALESCE(LIKES,0)) AS MIN, MAX(COALESCE(LIKES,0))AS MAX,AVG(COALESCE(LIKES,0))AS AVG  FROM TIP

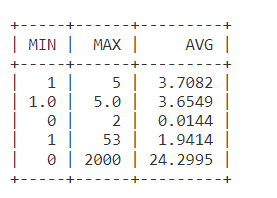
UNION ALL

 SELECT MIN(COALESCE(COUNT,0)) AS MIN, MAX(COALESCE(COUNT,0))AS MAX,AVG(COALESCE(COUNT,0))AS AVG FROM CHECKIN

UNION  ALL

 SELECT MIN(COALESCE(REVIEW\_COUNT,0)) AS MIN, MAX(COALESCE(REVIEW\_COUNT,0))AS MAX, AVG(COALESCE(REVIEW\_COUNT,0))AS AVG  FROM USER;

Output:



5. List the cities with the most reviews in descending order:

SQL code used to arrive at answer:

Code:

SELECT city,SUM(Review\_count) as total\_count

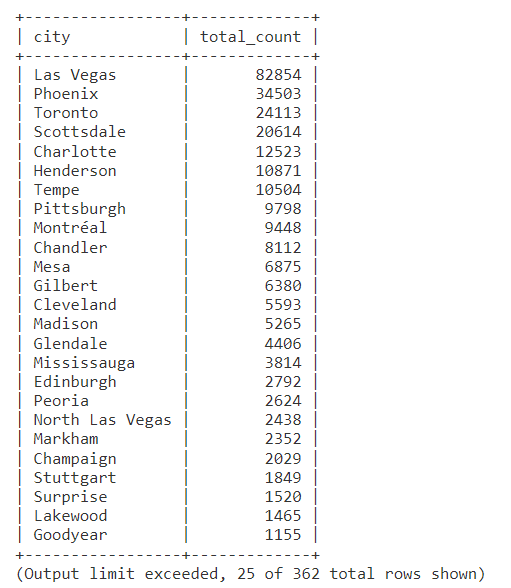
FROM business

GROUP BY city

ORDER BY total\_count desc;

Copy and Paste the Result Below:

**Output:**



6. Find the distribution of star ratings to the business in the following cities:

i. Avon

SQL code used to arrive at answer:

Code:

SELECT stars,count(stars)

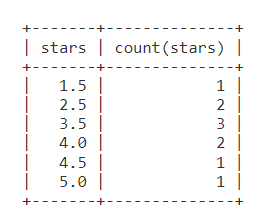
FROM business

WHERE city='Avon'

group by stars;

Copy and Paste the Resulting Table Below (2 columns â€“ star rating and count):

Output:



ii. Beachwood

SQL code used to arrive at answer:

Code:

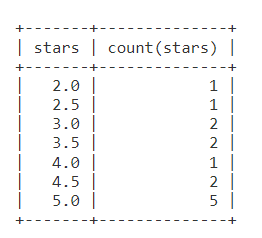
SELECT stars,count(stars)

FROM business

WHERE city='Beachwood'

group by stars;

Copy and Paste the Resulting Table Below (2 columns â€“ star rating and count):



7. Find the top 3 users based on their total number of reviews:

SQL code used to arrive at answer:

Code:

SELECT id,name, SUM(review\_count) AS total\_reviews

FROM user

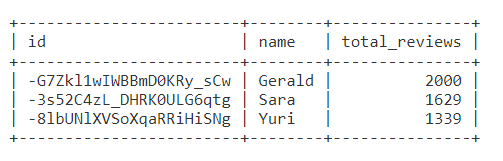
GROUP BY id

ORDER BY total\_reviews DESC

LIMIT 3;

Copy and Paste the Result Below:

Output:



8. Does posing more reviews correlate with more fans?

Please explain your findings and interpretation of the results:

Code:

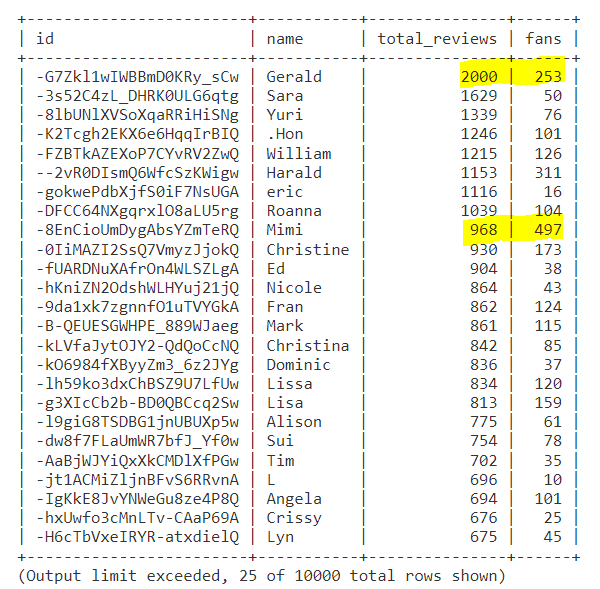
SELECT id,name,SUM(review\_count) AS total\_reviews,fans

FROM user

GROUP BY id

ORDER BY review\_count DESC;

Output:



Interpretation:

Posting more reviews does not correlate with more fans

Eg.Gerald posted 2000 reviews and have 253 fans whereas Mimi posted just 968 reviews but have 497 fans.

9. Are there more reviews with the word "love" or with the word "hate" in them?

Answer:Yes

SQL code used to arrive at answer:

Code:

SELECT

(SELECT COUNT(text)

FROM review

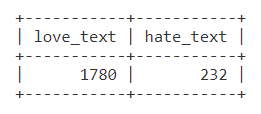
WHERE text LIKE "%love%") AS love\_text,

(SELECT COUNT(text)

FROM review

WHERE text LIKE "%hate%") AS hate\_text;

Output:



10. Find the top 10 users with the most fans:

SQL code used to arrive at answer:

Code:

SELECT id,name,SUM(fans) AS Total\_fans

FROM user

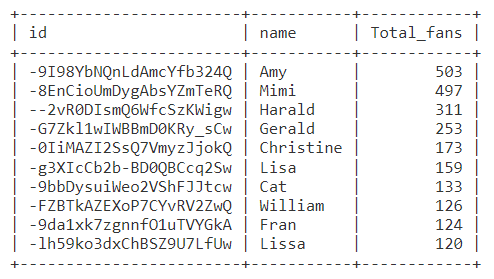
GROUP BY id

ORDER BY Total\_fans DESC

LIMIT 10;

Copy and Paste the Result Below:

Output:



Part 2: Inferences and Analysis

1. Pick one city and category of your choice and group the businesses in that city or category by their overall star rating. Compare the businesses with 2-3 stars to the businesses with 4-5 stars and answer the following questions. Include your code.

City=Toronto

Category=Restaurants

i. Do the two groups you chose to analyze have a different distribution of hours?

Yes

ii. Do the two groups you chose to analyze have a different number of reviews?

Yes

iii. Are you able to infer anything from the location data provided between these two groups? Explain.

The locations based on the postal codes are very different and do not correlate to ratings or reviews.

SQL code used for analysis:

Code:

SELECT b.name,b.stars,c.category,h.hours,b.review\_count,b.postal\_code

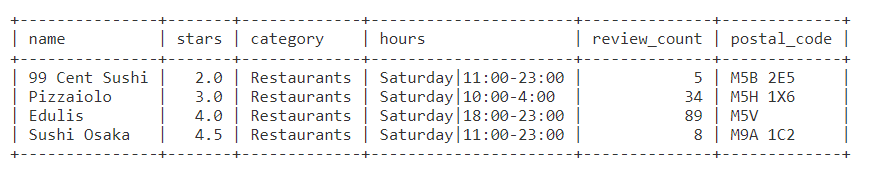
FROM (business b INNER JOIN category c ON b.id=c.business\_id)

INNER JOIN hours h ON  b.id=h.business\_id

WHERE city='Toronto' AND category='Restaurants'

GROUP BY b.stars;

Output:



2. Group business based on the ones that are open and the ones that are closed. What differences can you find between the ones that are still open and the ones that are closed? List at least two differences and the SQL code you used to arrive at your answer.

i. Difference 1:The businesses that are still open have a higher star rating.

ii. Difference 2:

The businesses that are still open have more reviews.

SQL code used for analysis:

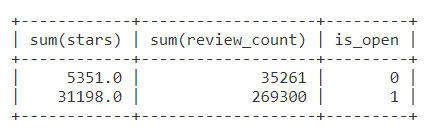
Code:

SELECT sum(stars),sum(review\_count),is\_open

FROM business

GROUP BY is\_open;

Output:



3. For this last part of your analysis, you are going to choose the type of analysis you want to conduct on the Yelp dataset and are going to prepare the data for analysis.

Ideas for analysis include: Parsing out keywords and business attributes for sentiment analysis, clustering businesses to find commonalities or anomalies between them, predicting the overall star rating for a business, predicting the number of fans a user will have, and so on. These are just a few examples to get you started, so feel free to be creative and come up with your own problem you want to solve. Provide answers, in-line, to all of the following:

i. Indicate the type of analysis you chose to do:

**The analysis is to find out what are the most successful businesses based on the business category.**

ii. Write 1-2 brief paragraphs on the type of data you will need for your analysis and why you chose that data:

**This analysis will require data from the business table and category table, such as id, stars, and review count.**

**The number of companies within each category, the average stars given by consumers, and the total reviews will need to be counted to determine if the data is relevant and unbiased.**

**To reduce irrelevant data, we're only looking at categories with at least 10 companies and an average of 3.0+ stars.**

iv. Provide the SQL code you used to create your final dataset:

Code:

SELECT  c.category,

        count(distinct b.id) AS num\_companies,

        round(avg(b.stars),2) AS avg\_stars,

        sum(b.review\_count) total\_reviews

FROM business b

JOIN category c ON b.id = c.business\_id

GROUP BY c.category

HAVING avg\_stars >= 3.0 AND num\_companies >= 10

ORDER BY avg\_stars DESC;

iii. Output of your finished dataset:

Output:

